Research Paper | Linguistics and Technology | E-ISSN No : 2454-9916 | Volume : 10 | Issue : 3 | March 2024

IMPLICATIONS OF SUMERIAN IN THE DIGITAL REVOLUTION INVOLVING CRYPTOGRAPHY, NLI AND MACHINE TRANSLATION

Sparsh Garg

United Arab Emirates

ABSTRACT

Sumerian is an extinct language whose potential has been hidden from us. This research talks about how the different features of Sumerian make it suitable to be used in cryptography, NLI and machine translation. In this research, we are trying to understand how an ancient language, that has been rendered useless for centuries, which has no significant use in today's society still serves as an excellent language model for the modern technological era. In this research we aim to showcase how Sumerian which has been silent through centuries, still has the potential to apprise and reinforce the understanding of past in a way that will help us mold the present according to our standards.

INTRODUCTION

The Mesopotamian civilization is one of the oldest civilizations to exist. The language used by the Mesopotamians; Sumerian is the oldest language to be deciphered till date. Although it is extinct, the language can have wide implications in the present digital age. Its features can be used heavily in computer science. The complex script has several benefits in making encryption, and NLPs more efficient and advanced. In this paper, we are going to explore the different features of Sumerian which makes it a very useful tool in today's world. Sumerian can be inculcated into Cryptography and NLI and Machine Translation. We will explore how languages that were used in the past, are still relevant in the modern era and are powerful enough to have a strong impact on the already prevailing ecosystem. The special features of Sumerian makes it a highly influential resource. The mixture of ancient knowledge into the current technology might be able to catalyze the growth process in a way we have never imagined.

CRYPTOGRAPHY

Sumerian Agglutinative Morphology in Cryptography:

Agglutinative morphology is type of morphological structure which involves formation of words by stringing together in morphemes in a fashion which is quite linear in nature. Each of these showcase a specific and unique grammatical or semantic meaning. Morphemes are quite easily separable in agglutinative languages. They also have a specific grammatical function. Agglutinative languages are essentially synthetic languages that incorporate the use of agglutination. A few examples of agglutinative languages are Turkish, Finnish, Japanese etc. Sumerian is also considered to be an agglutinative language. It has all the features of an agglutinative language. This long extinct language's ability of Agglutinative Morphology can be used to make cryptography, and encryption specifically, more efficient.

Cryptographic Key Generation and Regulation:

Agglutinative morphology can be used to create more protective and secure cryptographic keys. The use of the 'agglutinative language' properties of Sumerian might help in increasing the dependency on these keys. We could use the process of 'Root Morphemes' to achieve the goal. This process involves defining a certain set of root morphemes. These must represent and showcase the key properties. For instance, using the word 'shakash' which represents the function 'protect'. The language being extinct has 0 active speakers currently. The language was last used in the literary field in 100 A.D. Thus, using Sumerian root words is one of the safest bets involving very minimal risks. After defining the root word, the next step involves providing descriptive affixes. These affixes give more details regarding the function of the encrypted message. This makes the encryption of the function stronger. You may use either Sumerian or English words for these. The use of Sumerian as the root morpheme enables you to have many affixes. Agglutinative languages tend to have a wide range of vocabulary. It is estimated that the total number of words in Sumerian range from 600,000 to 2,000,000. After adding the affixes, the next step involves organizing all these words in a random order. Arranging it in a randomized order is a crucial step in maintaining security. Randomized order of the morphemes would increase the complexity of the code and would also provide strong resistance against cyber attack or any attempt of cyber invasions or breaches. You could use the following python code to randomly organize the affixes and the root word

```python
import random

# Define a list of example morphemes
morphemes = ["shir", "nam", "bar", "-cryption", "-sign", "-auth", "-master", "-gil", "-ish", "-zil"

def randomize_morphemes(morphemes):
  """
  Randomly organizes a list of morphemes.

  Args:
    morphemes: A list of strings representing morphemes.

  Returns:
    A string with the morphemes randomly ordered.
  """
  # Make a copy of the list to avoid modifying the original
  morphemes_copy = morphemes[:]
  random.shuffle(morphemes_copy)
  return " ".join(morphemes_copy)

# Generate a random key using the function
random_key = randomize_morphemes(morphemes)

print(f"Randomly generated key: {random_key}")
```

Note: It is recommended to use Sumerian for affixes as it would make it more secure and harder to crevice the encrypted function.

To make the key encryption more efficient, we can incorporate dynamic key updates. These involve adding affixes in accordance with relevant context. This helps in embedding the work of the function in the key itself. This ensures that the encrypted key along with being secure, is also productive in nature. Another update involves time-based updates. Time based affixes can be added in the key which would automatically update the key with respect to its defined schedule. Another inclusion might include the mnemonic keys. Mnemonic key involves mapping morphemes to syllables or words. For instance, assigning 'shir' to 'shield', 'bar-zil' to 'skyscraper', etc. This would make it easier for the user to understand and read the key. It also enhances the overall security.

Sumerian in Algorithm Design:

Sumerian could be used in styling different algorithms. It has impactful use in designing stream ciphers. Morpheme combinations can be used in generating unique and unpredictable keystreams for different combinations. The morpheme might convey a special transformation that would be chipped to the data stream. Proper organization of these morphemes can lead to a huge number of plausible transformations, hence making it extremely secure and difficult to break into. The use of Sumerian would also increase flexibility of the design.

NLI AND MACHINE TRANSLATION

Named Entity Recognition (NER):

One of the main functions of Sumerian in NER would involve training data. Sumerian can be used to create exclusive dictionaries and gazetteers. It can be used to compile lists of named entities that are known. This can be done through the cuneiform spellings and semantic information. This improves the efficiency of the system as it helps the model to identify and analyze similar entities that are present in texts that have never been seen. The language can also be used to leverage existing resources. Moreover, labelled data can be created for model training by annotating Sumerian texts.

Another function of Sumerian in NRE involves feature engineering. This would incorporate character-level features, morphological features, and contextual features. Analyzing individual morphemes, cuneiform combinations, and the surrounding semantic relationships of the words provides for a more secure NER model. Sumerian can be incorporated in already existing NER architectures. For instance, cuneiform system can be incorporated into Conditional Random Fields (CRF) or Long Short-Term Memory (LSTM) networks. Recently, there has been exploration of such models. The BERT transformer model, a pre-trained model was fine-tuned on a Sumerian corpora for NER tasks.

A better inculcation of the Sumerian systems would be in a graph neural network architecture as they would be able to map and capture the interconnected nature of entities better.

Machine Translation (MT):

The cuneiform system present in Sumerian has a logographic nature provides for unique combinations for specific entities which aids in recognition. This is more helpful than tradition alphabetical languages, which work on homophones. Moreover, the language provides for a controlled vocabulary. It has a very well-defined set of types of entity and domains. This creates a focused environment for training and yields a higher rate of accuracy in the categories it has been trained. Creating MT systems for Sumerian leads to the production of resources for new languages. These resources might include dictionaries, corporas, and other tools which provide linguistic benefits in research. Sumerian's incorporation in MT helps in pushing the boundaries for AI, making a more developed and holistic technological growth. It advances the machine translation capabilities. Sumerian MTs would also provide for multilingual NLP development.

Language Modelling:

The different language models might include Statistical Language Models (SLM), Neural Language Models (NLM), and Multilingual Language Models. Techniques like n-gram models help in understanding statistical relationships between different Sumerian texts. Furthermore, NLMs introduce computer architectures like LSTMs and transformers to capture the intricate relationships in Sumerian. The NLM model also helps in overcoming limitations of the SLM model. MLMs are a very useful tool. Using this model, relationships, and knowledge from other related languages such as Akkadian can also be leveraged. This further helps in enhancing modeling and communication capabilities.

CONCLUSION

The process of using Sumerian might face backlashes and hurdles. The extinct nature of the language can lead to several challenges. The presence of limited resources adds on to the ambiguity of the language itself. This also leads to confused use of homophones. Overall, Sumerian can be very beneficial in enhancing the security of the entire data transferring ecosystem and can also help in a more organized and developed artificial intelligence growth. Further research on its inculcation into cryptography, machine translations and NLI can contribute towards opening the doors of extensively using extinct languages in this field of computer science.

REFERENCES

1. Alkhzaimi, H. A. (2016). Cryptanalysis of selected block ciphers. Welcome to DTU Research Database. https://orbit.dtu.dk/en/publications/cryptanalysis-of-selected-block-ciphers
2. Borimori. (n.d.). English - to - Sumerian translator. Scribd. https://www.scribd.com/document/251055758/English-To-Sumerian-Translator
3. Chiarcos, C., Khait, I., Pagé-Perron, É., Schenk, N., Jayanth, Fäth, C., Steuer, J., Mcgrath, W., & Wang, J. (2018). Annotating a Low-Resource Language with LLOD Technology: Sumerian Morphology and Syntax. Information, 9(11), 290. https://doi.org/10.3390/info9110290
4. Luo, L., Liu, Y., Hearne, J., & Burkhart, C. (2015, April). Unsupervised sumerian personal name recognition. In The Twenty-Eighth International Flairs Conference.

5. Pearce, L. E. (1982). Cuneiform cryptography: numerical substitutions for syllabic and logographic signs. Yale University.
6. Prince, J. D. (1907). Sumerian as a Language. The American Journal of Semitic Languages and Literatures, 23(3), 202-219.
7. Bard, Google AI. "Potential uses of Sumerian in NLI and Machine Translation" Bard, Google AI, [2023-01-21]. Accessed [2023-01-21].